ÖREBRO UNIVERSITET

INSTITUTIONEN FÖR
NATURVETENSKAP OCH TEKNIK

Datorteknik för civilingenjörer, DT509G

Datorteknik, teori

A001
5 högskolepoäng

Skriftlig tentamen

2022-11-19

Programmering grundkurs, DT509G (A001)

Tillåtna hjälpmedel: penna, radergummi, engelska-svenska ordbok

Instruktioner:

- ***Läs igenom alla frågor noga.***
- **Ange tentamenskoden på svarsdokumentet.**
- **Du kan svara på *Svenska* eller *Engelska*.**
- ***Skriv tydligt* (gäller även för en digital tentamen).**
- **Detta är en individuell examination - alla misstankar om otillåtet samarbete kommer att rapporteras.**
- **Ansvarig lärare finns tillgänglig via telefon fr.o.m. andra skrivtimmen.**
- **Skriv läsligt!**
- **förklara och motivera era svar**

Ansvarig lärare: Pascal Rebreyend, tel: 0702001422

För betyg G krävs 50% av total poäng (20 på 40)
(26 poång gav betyg 4, och 33 poäng gav betyg 5)

Lycka till!

Question 1: (5 points)

You will find below a pseudo-code doing some computation on a bi-dimensional array (matrix). Can you find the best way to create the array p by allocating the memory (with malloc or calloc) and replace the line LA by the correct and corresponding C-code. We don't take into account how the array p is filled with its contents. Explain and motivate your choice.

```
double f1(int *p)
// p is a pointer to a bi-dimensional array of nb doubles. (a matrix)
{
  int max_raw=1000;  //number of raws in p
  int max_columns=10000; //number of columns in p
  int m1,m2;
  double a;
  a=0;
  for (m1=0;m1<max_raw;m1++)
    for (m2=0;m2<max_column;m2++)
          {
LA:       a=a+ <element of p at the column m2 and raw m1>;
          }
  return a;
}
```

Question 2: (2 points)

Why some processors have as instruction an arithmetic shift? In which context we may use it?

Question 3: (3 points)

In C we use often the struct constructor to create a new datatype.   Is the size of the whole struct (you can get it via the function sizeof()) the sum of the sizes of all its elements taken individually? (Justify)

Question 4: (3 points)

You are analyzing a assembly code produced by a C compiler and you have access to the original C source code. The compiler you are studying has two operation modes: one without any optimization at all and one with all optimizations implemented in the C compiler turned on. This C compiler is known to be really efficient and powerful since almost all known optimizations are implemented in it.
If you look on the two possible assembly code generated for the same source code, will be the order of (assembly) instructions the same or not? Why ?

Question 6: (4 points)

You are analyzing a software doing computations on matrices (bi-dimensional arrays).
First, you run twice the software with the same matrices of size N and the running time is 642 ms for the first run and 98 ms for the second run.
Then, you repeat the experiment with other matrices of size (N+10) (10 is really small in comparison of N) and you get a running time of 694 ms for the first run and 592ms for the second run. (both runs are also done with the same matrices as before).
Explain or give hypothesis why you have big difference in time for matrices of size N and for matrices of size N+1 the difference between the two consecutive runs is small. (For both sizes of matrices, experiments have been done in the same without any other software running on the computer).

Question 7: (2 points)

Why people often say that a write-through cache is more reliable than a write-back cache when the power of the system is unreliable? (in case the final destination is an Hard-Drive or a memory keeping information when power is lost like a usb-key)

Question 8: (2 points)

If you look on the different set of instructions for different cpus, you will find that depending the cpu the number of parameters differ. Often modern CPUs have 3 parameters (Example: ADD Ra,Rb,Rc which put in register Ra the result of the addition of values of Rb and Rc.)
Do we have CPU with less number of operands and how they work if as example we want to do an addition?

Question 9: (3 points)

On a modern CPU, operands of an instruction like the addition (assembly) can be of different types? What are or can be these types?

Question 10 : (4 points)

If you have a register (containing 32 bits) representing a number using the 2’s complement notation. Which instruction you will use to do a division by 2. (We assume the most significant bit is on the left)

Question 11: (7 points)

Write an ARM-like assembly code which perform the same task as the following C-Code:

```
int a,b,c,d;
a=7;    // in your solution 7 should be loaded in a register
b=0;
while (a>0)
  {
   b=b+a;
   c=a*b;
   for (d=4;d<c;c++)
        {
          b++;
        }
   a=a-1;
  }
```

Among all instructions, you can use:

- LD (load): Load an immediate value into a register
- ADD,MUL,SUB: addition, multiplication, substraction
- CMP: comparison between 2 registers
- BE,BNE: Branch if equal, Branch if not equal (based on the result of the previous CMP instruction)
- BG,BGE: same for branch if greater, branch if greater or equal
- JUMP: Jump to a label
- And other classical assembly instruction if needed

Question 12: (5 points)

Explain what is a buffer, how it works (a pseudo-code in a C-like language would be appreciated) and why we often use buffer?
Describe also disavantages and inconveniences of a buffer.